NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2004; **11**:773–793 (DOI: 10.1002/nla.346)

Matrix-free constructions of circulant and block circulant preconditioners

Chao Yang[1,*,†], Esmond G. Ng[1] and Pawel A. Penczek[2,‡]

[1]*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50F-1650, Berkeley, CA 94720-8139, U.S.A.*
[2]*Department of Biochemistry & Molecular Biology, The University of Texas, Houston Medical School, Houston, TX 77030, U.S.A.*

SUMMARY

A framework for constructing circulant and block circulant preconditioners (C) for a symmetric linear system $Ax = b$ arising from signal and image processing applications is presented in this paper. The proposed scheme does not make explicit use of matrix elements of A. It is ideal for applications in which A only exists in the form of a matrix vector multiplication routine, and in which the process of extracting matrix elements of A is costly. The proposed algorithm takes advantage of the fact that for many linear systems arising from signal or image processing applications, eigenvectors of A can be well represented by a small number of Fourier modes. Therefore, the construction of C can be carried out in the frequency domain by carefully choosing the eigenvalues of C so that the condition number of $C^{\mathrm{T}}AC$ can be reduced significantly. We illustrate how to construct the spectrum of C in a way that allows the smallest eigenvalues of $C^{\mathrm{T}}AC$ to overlap with those of A extremely well while making the largest eigenvalues of $C^{\mathrm{T}}AC$ several orders of magnitude smaller than those of A. Numerical examples are provided to demonstrate the effectiveness of the preconditioner on accelerating the solution of linear systems arising from image reconstruction applications. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: iterative methods; Toeplitz matrices; circulant preconditioners; frequency response

1. INTRODUCTION

Circulant and block circulant preconditioners have been proposed as potential candidates for accelerating iterative solutions of Toeplitz or block Toeplitz systems

$$Ax = b \tag{1}$$

*Correspondence to: Chao Yang, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50F-1650, Berkeley, CA 94720-8139, U.S.A.
†E-mail: cyang@lbl.gov
‡E-mail: Pawel.A.Penczek@uth.tmc.edu

Contract/grant sponsor: U.S. Department of Energy; contract/grant number: DE-AC03-76SF00098
Contract/grant sponsor: National Institute of Health; contract/grant number: R0I GM60635
Contract/grant sponsor: National Energy Research Scientific Computing Center

Received 31 October 2001
Published online 17 May 2004 *Revised 23 July 2002*
 Accepted 24 July 2002

These systems often arise from signal and image processing applications. The algorithm proposed by Chan [1] constructs a circulant preconditioner C that minimizes $\|A - C\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. The same technique can be extended to construct block circulant matrices with circulant blocks (BCCB) for block Toeplitz matrices with Toeplitz blocks (BTTB) [2]. Other possible constructions have been presented in References [3–7]. Almost all of these techniques rely heavily on the Toeplitz structure of A. Although it has been shown that the technique of Chan [1] can be applied to matrices that are nearly Toeplitz or non-Toeplitz [8], one must know all entries of A in order to construct C.

The spectral properties of circulant preconditioners have been analysed by Chan and others [9–11] in the context of Toeplitz systems in which the matrix entries of A are assumed to be Fourier coefficients of a *generating* function of a certain class (e.g. the Wiener class). Similar analysis for block circulant matrices used in Toeplitz-block least squares problems are provided in References [12, 13]. Construction techniques based on polynomial approximation of a generating function is given in Reference [14]. In many applications, such a generating function does not exist or is unknown *a priori*.

In this paper, we propose a framework for constructing circulant and block circulant preconditioners of A without making explicit use of the entries of A. This approach is extremely useful for applications in which the matrix A is not explicitly available, but exists, for example, in the form of a matrix–vector multiplication routine. One particular example is the matrix operator associated with real space image reconstruction from a set of projections [15]. In this case, A is dense but not Toeplitz or block Toeplitz. Each column (or row) of A corresponds to a spatially variant point spread function (PSF). The operation $y \leftarrow Ax$ represents a combined discrete projection and back-projection operation. Although the entries of A can be computed from the transforms of point sources, it is generally too costly to store all entries of A in memory. For example, to reconstruct a $64 \times 64 \times 64$ 3-D image, we would work with a matrix A that is $2^{18} \times 2^{18}$, containing roughly 32 000 000 000 non-zero entries. (Since the matrix is symmetric, we only counted the lower half of the matrix.) If each entry is represented by a 4-byte floating point number, this would lead to a 128 gigabyte (GB) storage requirement.

Our approach is motivated by the observation that in these applications eigenvectors of A can often be well represented by a small number of Fourier modes. To be specific, the magnitude of the discrete Fourier transform (DFT) of each eigenvector often exhibits a localized peak.

To give an example, let us take A to be a modified version of the `Phillips` matrix collected in Reference [16]. The dimension of the matrix is set to $n = 128$. The original matrix is a Toeplitz matrix obtained from a discretization of a Fredholm integral equation of the first kind [17]. Our modification reverses the sign of the negative eigenvalues while keeping the eigenvectors unchanged. As a result, the modified matrix is no longer Toeplitz. This simple modification makes A symmetric and positive definite (SPD) so that the conjugate gradient (CG) algorithm may be applied later on to solve (1).

We assume that eigenvalues of A have been arranged in descending order $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, and z_i is the eigenvector associated with λ_i. In Figure 1, we plot the eigenvectors associated with λ_1, λ_{64}, λ_{128} and the frequency content defined by the magnitude of the DFT of each vector. The DFT of a vector x, often denoted in this paper by $y = \mathrm{DFT}\{x\}$, can be expressed, in matrix notation, by

$$y = F^{\mathrm{H}} x$$

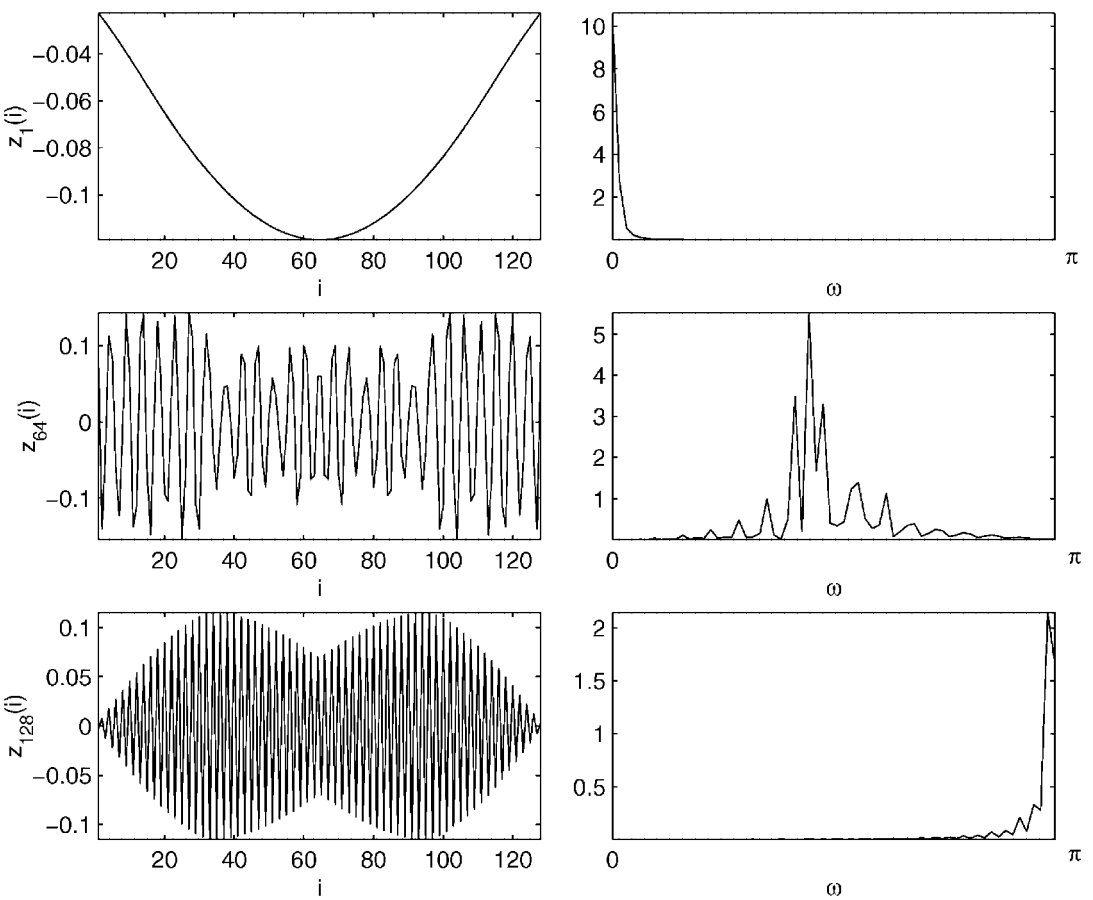

Figure 1. Eigenvectors and their Fourier frequency content.

where

$$F = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega_0^0 & \omega_0^1 & \cdots & \omega_0^{n-1} \\ \omega_1^0 & \omega_1^1 & \ddots & \omega_1^{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ \omega_{n-1}^0 & \omega_{n-1}^1 & \cdots & \omega_{n-1}^{n-1} \end{bmatrix} \quad \text{and} \quad \omega_i^k = \mathrm{e}^{-\mathrm{j}(2\pi/n)ik}$$

The superscript H is used here to denote conjugate transpose. We have adopted the engineering convention of using the letter j to denote the imaginary constant $\sqrt{-1}$ above. The frequency content computed by DFT is normally displayed on $[0, 2\pi]$. However, due to the symmetric pattern of each eigenvector in this example, the discrete Fourier coefficients are even symmetric with respect to $\omega = \pi$ (the *Nyquist* limit). Thus we only plotted the portion on $[0, \pi]$.

Clearly, z_1 contains mainly the low frequency Fourier components, and z_{128} the high frequency components. This type of correspondence between the large (small) eigenvalues of A and the low (high) Fourier frequencies is typical for discretized integral operators. The eigenvector z_{64} associated with the interior eigenvalue, λ_{64}, contains contributions from a larger number of Fourier modes. But the localization of Fourier components is still visible.

If the eigenvectors of A were indeed columns of F, then it is easy to show that A is circulant and eigenvalues of A are determined by the first column of the matrix [18]. In this case, the ideal preconditioner C would be circulant as well. The first column of C (which ultimately defines the entire matrix) can be obtained by taking the (inverse) DFT of the 'inverse' frequency response $[1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_n]$. Of course, when A is circulant, one should not use iterative method to solve (1). A simple inversion algorithm based on the fast Fourier transform (FFT) can be applied directly to render the solution in $\mathcal{O}(n \log n)$ flops. Note that, in system theory, eigenvalues of a circulant matrix are sometimes known as the frequency response of the linear convolution operator defined by the matrix.

The simple procedure described above cannot be used in general to construct a preconditioner for a non-circulant matrix A for which the spectrum is not readily available. However, the close connection between the eigenvectors of A and the Fourier modes would allow us to choose a C such that $\|CAx\| \leqslant \|Ax\|$, for all x such that $\|x\| = 1$. In particular, if A is a discretized integral operator, and the eigenvalues of C associated with the lowest frequency Fourier modes are chosen appropriately, it is possible to achieve

$$\lambda_n \leqslant \max_{x,\,\|x\|=1} \|CAx\| < \max_{x,\,\|x\|=1} \|Ax\| = \lambda_1$$

Meanwhile, by making the eigenvalues of C associated with the high frequency Fourier modes close to 1, we can make the smallest eigenvalues of CA overlap with those of A. Consequently, the condition number of the preconditioned system can be reduced.

To maintain symmetry of the coefficient matrix in (1), we prefer to work with

$$C^{\mathrm{T}}AC\hat{x} = C^{\mathrm{T}}b$$

where $x = C\hat{x}$ is the solution to the original problem (1).

The general strategy we take here is to carefully construct the spectrum (frequency response) of C so that eigenvalues of $C^{\mathrm{T}}AC$ are clustered. As mentioned earlier, the matrix C is completely defined once its spectrum is specified. If A is real, it is desirable to keep C real as well. In Section 2, we will show how a real circulant preconditioner can be constructed when the amplitude of the desired frequency response is available. Most of the materials presented there can be found in standard literature on digital filter design [19, 20]. We include them here merely for completion. In Section 3, we will illustrate how to devise a frequency response that leads to an effective circulant preconditioner. The discussions presented in Sections 2 and 3 focus on one dimensional (1-D) problems. These techniques can be easily extended to 2 or 3-D problems as we will show in Section 4.

2. CIRCULANT MATRIX AND FREQUENCY RESPONSE

A circulant matrix

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & \ddots & c_1 & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}$$

is completely specified by its first column, $c = [c_0, c_1, \ldots, c_{n-1}]^{\mathrm{T}}$. It is well known that

$$C = F \Lambda F^{\mathrm{H}} \tag{2}$$

where F is the DFT matrix defined in Section 1, and Λ is a diagonal matrix with the diagonal entries defined by the DFT of c, i.e.

$$\mathrm{diag}(\Lambda) = \mathrm{DFT}\{c\} = F^{\mathrm{H}} c$$

The matrix–vector multiplication $y = Cx$ is essentially a circular convolution

$$y_k = \sum_{i=0}^{n-1} c_{i-k} x_i, \quad k = 0, \ldots, n-1$$

where $c_{i-k} = c_{\mathrm{mod}(i-k,n)}$ for $i - k < 0$. In system theory, the discrete-time Fourier transform (DTFT) [19] of c is also known as the frequency response of the space-invariant linear system characterized by the impulse response $\{c_0, c_1, \ldots, c_{n-1}\}$. It follows from the spectral decomposition (2) that $y \leftarrow Cx$ can be carried out by the following steps:

1. $f \leftarrow \mathrm{DFT}\{x\}$;
2. $f \leftarrow \Lambda \cdot f$;
3. $y \leftarrow \mathrm{IDFT}\{f\}$;

In step 1, the input vector x is decomposed as a linear combination of the Fourier basis vectors, the Fourier coefficients contained in f are then point-wise modified by the frequency response of the linear system before they are reassembled by the inverse DFT (IDFT) to produce the output vector y.

Because DFT can often be implemented efficiently by making use of the fast Fourier transform (FFT), the computational complexity of $y \leftarrow Cx$ can be reduced to $O(n \log n)$ from $O(n^2)$.

To make our discussion easier, we introduce the following elementary properties of DFT and convolution. In particular, we point out the implication of symmetry in the impulse response $\{c_i\}$.

2.1. Linear phase and symmetric impulse response

Our construction of a circulant preconditioner begins with a specification of the amplitude of the desired frequency response $a(\omega_k)$ at $\omega_k = (2\pi/n)k$, $k = 0, 1, \ldots, n-1$. After appropriate phase components $\phi(\omega_k)$ are selected, the impulse response c_i required to assemble C can be obtained from the inverse DFT of $f(\omega_k) = a(\omega_k)\mathrm{e}^{\phi(\omega_k)}$.

Since the matrix A in (1) is real, we would like C to be real as well. This suggests that the phase components of the frequency response cannot be arbitrary. For example, setting $\phi(\omega_k) = 0$ for all k will most likely result in an impulse response that contains a non-zero imaginary part.

A simple way to specify the phase content of the frequency response is to resort to the technique used in linear phase digital filter design [20]. If we let $m = (n-1)/2$, it follows from the DTFT [19] of $c = [c_0, c_1, \ldots, c_{n-1}]$ that

$$f(\omega) = \sum_{\ell=0}^{n-1} (\mathrm{e}^{-\mathrm{j}\omega})^{\ell} c_{\ell} \tag{3}$$

$$= \mathrm{e}^{-\mathrm{j}\omega m} \sum_{\ell=0}^{n-1} (\mathrm{e}^{\mathrm{j}\omega})^{m-\ell} c_{\ell} \tag{4}$$

Note that $f(\omega)$ is periodic with period $T = 2\pi$. Thus we only consider $f(\omega)$ defined on one period $[-\pi, \pi]$ or $[0, 2\pi]$, whichever is more convenient to work with.

It is easy to verify that

$$f(\omega) = a(\omega)\mathrm{e}^{-\mathrm{j}\omega m}$$

where

$$\begin{aligned} a(\omega) = {} & (c_0 + c_{n-1})\cos(m\omega) + \mathrm{j}(c_0 - c_{n-1})\sin(m\omega) \\ & + (c_1 + c_{n-2})\cos((m-1)\omega) + \mathrm{j}(c_1 - c_{n-2})\sin((m-1)\omega) \\ & + \cdots \end{aligned} \tag{5}$$

If we impose symmetry on the impulse response by setting

$$c_0 = c_{n-1}, \quad c_1 = c_{n-2}, \ldots$$

$a(\omega)$ becomes completely real. It forms the amplitude of the frequency response $f(\omega)$ whose phase $\phi(\omega) = m\omega$ is linear in ω. It is easy to see that $a(\omega)$ is even symmetric, i.e. $a(\omega) = a(-\omega)$ for $\omega \in [-\pi, \pi]$, a property desired in our construction.

It follows from (5) that, if n is odd, the impulse response can be calculated from the amplitude of the frequency response sampled on $[0, \pi]$ by solving a linear system of equations (if the number of sampling points equals the $m+1$) or a linear least squares problem (if the number of sampling points is greater than $m+1$). Suppose the frequency response is sampled

uniformly at $\omega_k = (2\pi/n)k$, $k = 0, 1, \ldots, m$, then

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 2\cos(m\omega_0) & 2\cos((m-1)\omega_0) & \cdots & 2\cos\omega_0 & 1 \\ 2\cos(m\omega_1) & 2\cos((m-1)\omega_1) & \ddots & \vdots & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 2\cos(m\omega_{m-2}) & \cdots & \ddots & 2\cos\omega_{m-2} & 1 \\ 2\cos(m\omega_{m-1}) & 2\cos((m-1)\omega_{m-1}) & \cdots & 2\cos\omega_{m-1} & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix} \tag{6}$$

where $a_k = a(\omega_k)$. A similar equation can be derived for an even n. It is easy to verify that the inverse of the coefficient matrix in (6) has the form

$$G = \frac{1}{n} \begin{bmatrix} 1 & 2\cos(m\omega_1) & \cdots & 2\cos(m\omega_{m-1}) & 2\cos(m\omega_m) \\ 1 & 2\cos((m-1)\omega_1) & \ddots & \vdots & 2\cos((m-1)\omega_m) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & \cdots & \ddots & 2\cos\omega_{m-1} & 2\cos\omega_m \\ 1 & 2 & \cdots & 2 & 2 \end{bmatrix}$$

Thus, the impulse response can be computed by a matrix–vector multiplication. An alternative (and perhaps more efficient) way of calculating the impulse response is to form the frequency response $f(\omega_k)$ by multiplying the desired amplitude samples a_k with the correct phase components $e^{m\omega_k}$, $k = 0, 1, \ldots, n-1$. An inverse FFT of $f(\omega_k)$ will yield the desired impulse response. A MATLAB script is provided in Figure 2 to demonstrate how this is done. Note that the `ifft(fq)` command will return a complex vector with tiny imaginary parts. In exact arithmetic, the imaginary part should be identically zero. Only the real part is of interest.

2.2. *Sparsity and frequency interpolation*

The even symmetry of the frequency response suggests that roughly $n/2$ non-zero entries are required to assemble the desired circulant preconditioner. If the amplitude of the desired frequency is smooth and well behaved (in the sense of having uniformly bounded derivatives) on $[0, \pi]$, the number of non-zeros in the impulse response can be further reduced, making the construction of circulant matrix even more efficient.

Suppose an impulse response of length p

$$\tilde{c} = [c_0, c_1, \ldots, c_{p-1}]$$

has been computed from the amplitude of p ($p \ll n$) uniformly sampled frequency response on $[0, \pi]$. An extended impulse response formed by padding $\tilde{c}$ with $n - p$ zeros at the end,


```
% assume n is odd, and
% f(x) is the amplitude of the desired
% frequency response;
%
m   = (n-1)/2;
j   = sqrt(-1);
h   = 2*pi/n;
x   = 0:h:2*pi-h;
y   = f(x(1:m+1));
yl  = [y y(m:-1:1)];
phs = exp(2*pi*j*(-m)*(0:n-1)/n);
fq  = yl.*phs;
cc  = ifft(fq);
c   = real(cc);
```

Figure 2. Calculating the impulse response by `ifft`.

i.e.

$$c = [c_0, c_1, \ldots, c_{p-1}, 0, 0, \ldots, 0]$$

produces a frequency response whose amplitude agrees with that of $\tilde{c}$ at $\omega_k = (2\pi/p)k$, $k = 0, 1, \ldots, p-1$. Figure 3 demonstrates this well-known interpolation property (*sinc* interpolation [19]). The circles mark the amplitude of the frequency response associated with a length-21 impulse response $\tilde{c}$. The solid curve corresponds to the amplitude of the frequency response computed from the length-128 impulse response formed by padding $\tilde{c}$ with 107 zeros at the end. Note that the frequency responses are computed by the MATLAB `fft` command. The amplitude curves are shifted (by making use of the MATLAB `fftshift` command) to the interval $[-\pi, \pi]$ before they are plotted.

The sparsity of C due to zero padding allows us to apply the preconditioner directly in the spatial domain through sparse matrix–vector multiplication. When $p \ll n$, the complexity of this approach is reduced to $O(n)$ from the familiar $O(n \log(n))$ count associated with the FFT algorithm.

2.3. Connection with polynomial approximation

The construction technique presented in Section 2.1 essentially provides a means to solve a trigonometric interpolation problem, i.e. to find α_ℓ, $\ell = 0, 1, \ldots, m$ such that

$$a(\omega_k) = \sum_{\ell=0}^{m} \alpha_\ell \cos((m-\ell)\omega_k) \tag{7}$$

at $\omega_k = (2\pi/n)k$, $k = 0, 1, \ldots, m$. If we let $\omega = \cos^{-1} x$, where $x \in [-1, 1]$ and ω is restricted to $[-\pi, \pi]$, it follows from the definition of Chebyshev polynomial of the first kind [21] that the amplitude of the frequency response can be expressed as a sum of Chebyshev basis

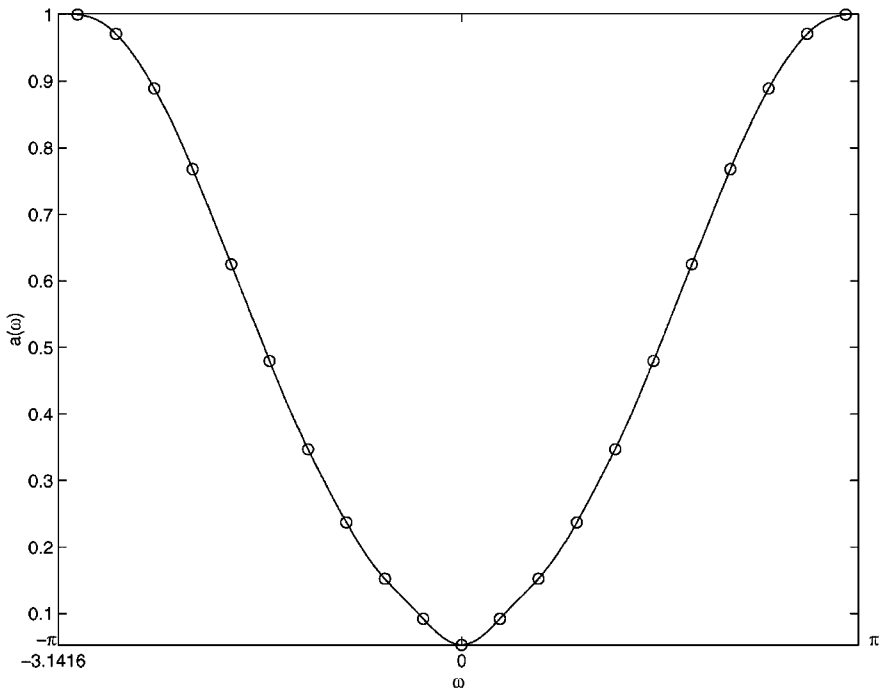

Figure 3. The interpolated frequency response.

polynomials defined on $[-1, 1]$, i.e.

$$a(x) = \sum_{\ell=0}^{m} \alpha_\ell T_{m-\ell}(x)$$

where $T_\ell(x) = \cos(\ell \cos^{-1}(x))$. The polynomial approximation to the desired frequency response is solved by interpolating at a set of Chebyshev points $x_k = \cos(2\pi/n)k$, $k = 0, 1, \ldots, m$. The length of the impulse response $n = 2m+1$ is proportional to the degree of the polynomial. When the amplitude of the desired frequency response is smooth and well behaved (in the sense of having uniformly bounded derivatives), a low-degree interpolating polynomial is often adequate to provide an accurate approximation. Otherwise, other approximation techniques such as weighted least squares and minmax approximation [21] may be used to provide a satisfactory impulse response. The degree of the polynomial may also be increased leading to a denser circulant preconditioner.

3. SQUEEZING THE SPECTRUM BY AN EXPONENTIAL FREQUENCY RESPONSE

In the previous section, we laid out a numerical procedure for constructing a circulant preconditioner when the amplitude of a desired frequency response is provided. In this section, we discuss how to come up with an appropriate frequency response to achieve the goal of reducing the condition number of A.

If we consider the spectrum of A as a sequence of real numbers (arranged in descending order) obtained from a continuous function $\lambda(\omega)$ uniformly sampled on the interval $[0,\pi]$, then a desirable property of the frequency response $f(\omega)$ associated with C is

$$f(\omega_k)=\frac{1}{\sqrt{\lambda_k}}$$

for $k=1,2,\ldots,n$. This property will potentially allow us to make the spectrum of $C^{\mathrm{T}}AC$ clustered around 1. However, since the spectrum of A is usually unknown, such a frequency response is not easy to construct. Nevertheless, it is possible to construct a monotonically increasing $f(\omega)$ such that $\lambda_n/\lambda_1 < f^2(\omega_k) < 1$ for $k \ll n$ and $f^2(\omega_k)\approx 1$, for $k \approx n$. As a result, the function $g(\omega)=f^2(\omega)\lambda(\omega)$, which can be viewed approximately as the 'frequency response' of the preconditioned matrix $C^{\mathrm{T}}AC$, may satisfy the property that

$$\frac{g(0)}{g(\pi)} < \kappa(A)=\frac{\lambda_1}{\lambda_n}$$

The ratio $\tau=f(\pi)/f(0)$ will be referred to as the *reduction factor* (of the condition number) in the following. Clearly, we would like to construct $f(\omega)$ with a large reduction factor τ. But τ should also be controlled not to exceed $\sqrt{\kappa(A)}$.

There are many ways to construct an $f(\omega)$ that satisfies the above requirements. One possible construction is to let

$$f(\omega)=\mathrm{e}^{-\beta(\omega-\pi)^k} \tag{8}$$

This exponential response has a number of interesting properties. It is easy to verify that

$$f(\pi)=1 \quad \text{and} \quad f^{(i)}(\pi)=0 \quad i=1,2,\ldots,k-1$$

where $f^{(i)}$ denotes the ith derivative of $f(\omega)$. For an even integer k and positive β, $f(\omega)$ increases monotonically on $[0,\pi]$. The larger the value of k, the flatter $f(\omega)$ is at $\omega=\pi$. The parameter β can be completely determined once k and the reduction factor τ are specified, i.e. we can find β by solving

$$\frac{1}{\tau}=\mathrm{e}^{-\beta\pi^k}$$

A simple algebraic manipulation yields

$$\beta=\frac{\ln\,\tau}{\pi^k} \tag{9}$$

Let us now illustrate the effect of such a construction using the modified `Phillips` matrix given in Section 1 as an example. The condition number of A is $\kappa(A)\approx 7\times 10^6$. The spectrum of A is plotted in Figure 4. Using $k=2$, $\tau=100$, we obtain $\beta=0.482$ from (9).

Once the desired frequency response is specified, we can use the interpolation techniques introduced in Section 2.1 to construct an impulse response that defines our circulant preconditioner C_e.

In Figure 5, we plot $f(\omega)=\mathrm{e}^{-0.482(\omega-\pi)^2}$ (the solid curve) and the amplitude of the frequency response (marked by circles) defined by the a length-21 impulse response computed by the procedure shown in Figure 2. Because the $f(\omega)$ is smooth and well behaved ($|f'(\omega)|<2$

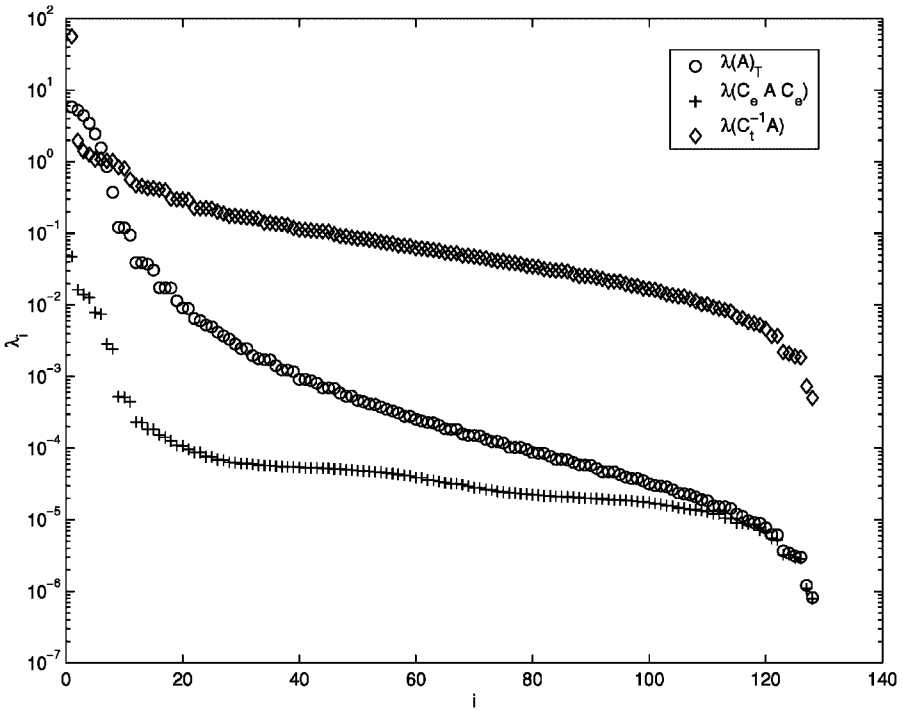

Figure 4. The spectrum of A and the preconditioned matrices.

for $\omega \in [0, \pi]$), the difference between the amplitude of the frequency response associated with $[c_0, c_1, \ldots, 7c_{21}]$ and $f(\omega)$ is negligible.

The spectrum of $C_e^{\mathrm{T}} A C_e$ (marked by plus signs) is compared with that of A (marked by circles) in Figure 4. Eigenvalues are sorted in descending order. In the same figure, we also plotted the spectrum of $C_t^{-1} A$, where C_t is the circulant preconditioner constructed using Chan's technique [1]. Notice that the smallest eigenvalues of $C_e^{\mathrm{T}} A C_e$ overlap extremely well with those of A. The largest eigenvalues of $C_e^{\mathrm{T}} A C_e$ are much smaller than those of A. That is, through preconditioning, eigenvalues of the preconditioned system has been squeezed into a much smaller interval. Consequently, the condition number of $C_e^{\mathrm{T}} A C_e$ is nearly four orders of magnitude smaller than that of A. This is consistent with the reduction factor $\tau = 100$ we specified for the construction. Chan's preconditioner reduced the condition number by nearly two orders magnitude also. The spectrum of the preconditioned matrix is less predictable, with most of the eigenvalues lie in $[10^{-3}, 1]$ and one outlier near 10^2.

In Figure 6, we examine the spectrum of $C_e^{\mathrm{T}} A C_e$, where C_e is constructed by setting the reduction factor $\tau = 10^3$ in (9). It is interesting to see that most of the eigenvalues of $C_e^{\mathrm{T}} A C_e$ have been squeezed into the interval $[10^{-4}, 10^{-6}]$ with one outlier near 10^{-2}.

The effect of the preconditioner in a preconditioned CG (PCG) iteration is shown in Figure 7. The right-hand side b used in this example is obtained from $b = Ax_e$, where A is our modified (SPD) `Phillips` matrix and x_e is an exact solution provided in Reference [16].

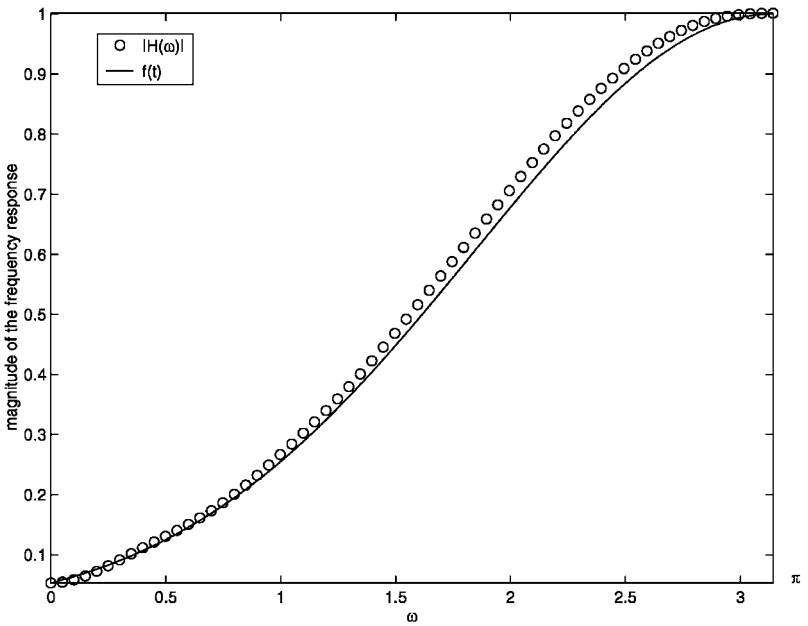

Figure 5. Exact and approximate frequency response.

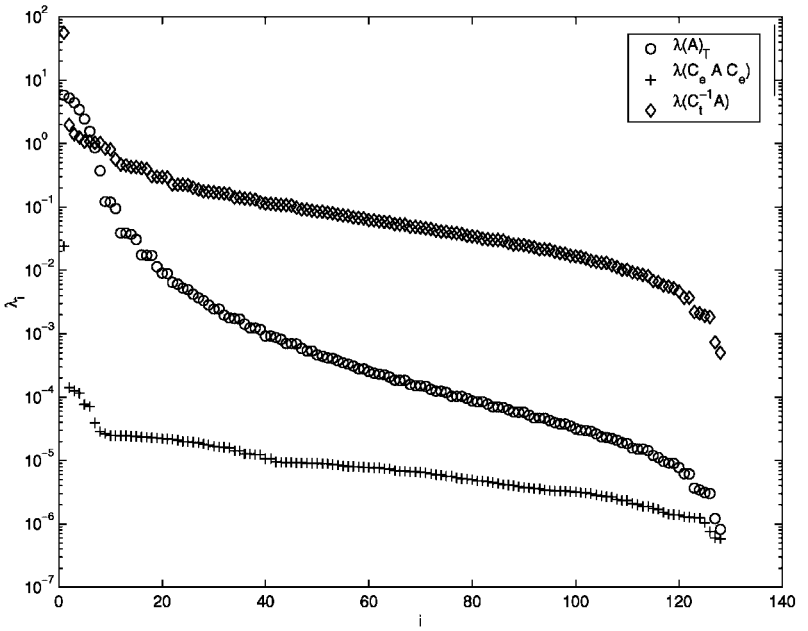

Figure 6. Spectrum of A and the preconditioned matrices, with a different τ for constructing $C_e^{\mathrm{T}} A C_e$.

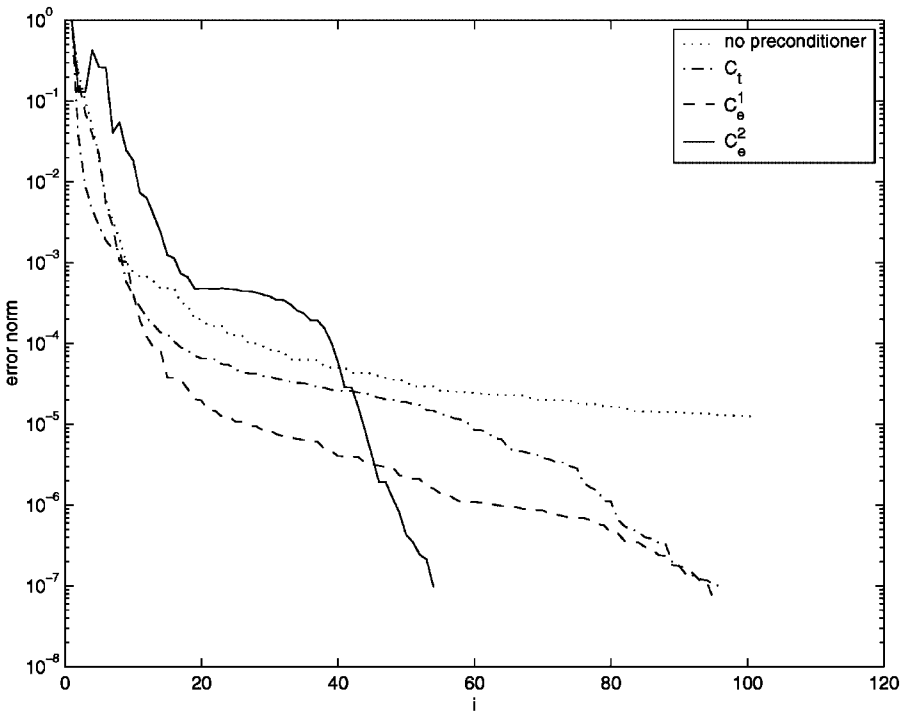

Figure 7. Convergence history of PCG.

We measure the convergence by examining the relative error defined by

$$\text{err} = \frac{\|x - x_e\|}{\|x_e\|} \tag{10}$$

We note that the convergence of PCG depends on both the spectral property of A and the right-hand side b. (A zero starting vector is used throughout this paper.) The convergence tolerance of PCG is set to $\text{tol} = 10^{-7}$, i.e. PCG is terminated when $\text{err} \leqslant 10^{-7}$.

The dotted curve in Figure 7 illustrates the convergence history of CG without using a preconditioner. The reduction in relative error starts to stagnate when the iteration number reaches 40. The Chan preconditioner (marked by the dash–dotted curve) appears to be superior in the first 10 iterations. However, nearly 100 iterations are required before the relative error of the solution falls below the required tolerance. Similar behaviour is observed for the matrix-free circulant preconditioner $C_e^{\tau_1}$ constructed by setting the reduction factor τ_1 to 100. The preconditioner $C_e^{\tau_2}$ associated with a reduction factor of $\tau_2 = 1000$ appears to be much more effective after the initial 40 iterations.

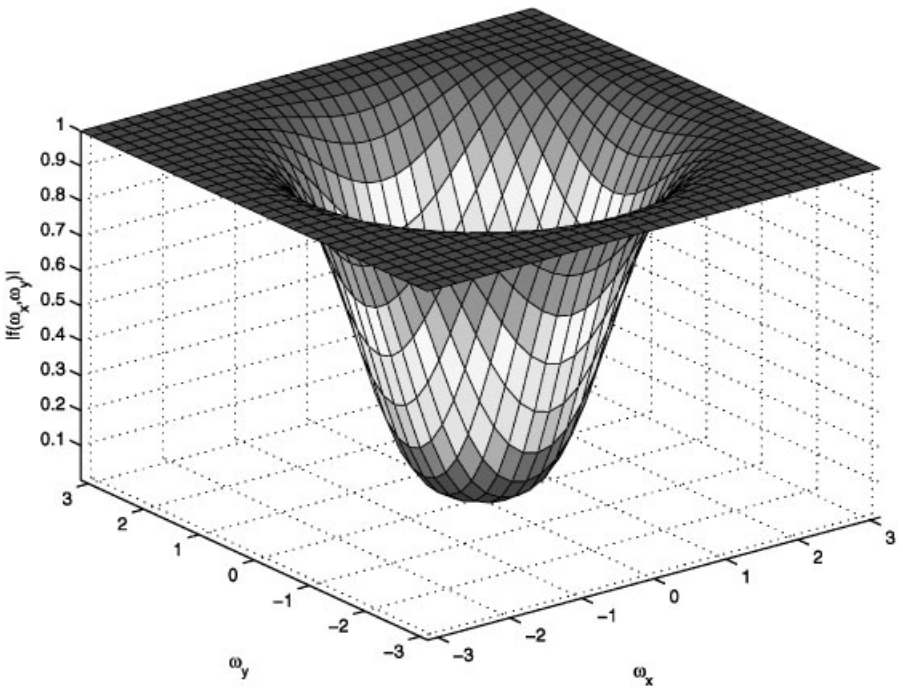

Figure 8. The amplitude of a desired 2-D frequency response.

4. BLOCK CIRCULANT PRECONDITIONERS

The construction techniques introduced above can be easily extended to block matrices for which eigenvectors can be well represented by a small number of multi-dimensional Fourier vectors. (The 2-D Fourier vectors form the columns of the matrix $F \otimes F$, where $\otimes$ denotes the Kronecker product.) Examples of these matrices are BTTB matrices (Block Toeplitz matrices with Toeplitz Blocks) and discrete image blurring operator associated with spatially variant point spread functions.

What is required for the construction of a BCCB matrix (block circulant matrix with circulant blocks) is the specification of a 2-D frequency response. This can be obtained by simply rotating the amplitude of a 1-D frequency response defined in (8) around 0 in the 2-D frequency domain. An example of such a frequency response is shown in Figure 8 where the frequency domain is restricted to $[-\pi, \pi] \times [-\pi, \pi]$. Notice that the frequency surface takes the value 1 in the high frequency region (four corners and boundary). The flatness of the surface can be controlled by the value of k in (8). Making the high frequency response surface flat has the effect of keeping the smallest eigenvalue of $C_e^{\mathrm{T}} A C_e$ in roughly the same location as those of A. The magnitude of the frequency response at $(0, 0)$ is determined by the reduction factor τ as described in Section 3 for the 1-D case. By using a large τ, we hope to push the largest eigenvalues of $C_e^{\mathrm{T}} A C_e$ toward the smallest eigenvalues, thereby reducing the condition number of the linear system.


```
% assume p is odd, and
% f(x) is the amplitude of the desired
% frequency response on [0,pi]x[0,pi];
%
m = (p-1)/2;
j = sqrt(-1);
h = 2*pi/p;
x = 0:h:2*pi-h;
y = x;
[X,Y] = meshgrid(x,y);
F     = zeros(p)
R     = sqrt(X(1:m+1,1:m+1).^2+Y(1:m+1,1:m+1).^2);
Z     = f(R);
F(1:m+1   ,1:m+1)    = Z;
F(p:-1:m+2,p:-1:m+2) = Z(2:m+1,2:m+1);
F(1:m+1   ,p:-1:m+2) = Z(:    ,2:m+1);
F(p:-1:m+2,1:m+1)    = Z(2:m+1,:);
phs   = exp(2*pi*j*(-m)*(0:p-1)/p);
PHS   = phs.'*phs;
FQ    = F.*PHS;
c2    = real(ifft2(FQ));
```

Figure 9. Calculating the 2-D impulse response by `ifft2`.

The procedure for calculating the 2-D impulse response from a desired frequency response is illustrated in Figure 9. Just as in the 1-D case, the value of p can be chosen to be much smaller than the block size of A. The computed 2-D impulse response can be padded with zeros to yield a sparse BCCB preconditioner. The interpolation property described in Section 2.3 also holds here.

The effect of block circulant on the spectrum of A can be seen in Figure 10. The matrix A used here is the `blur` matrix collected in Reference [16]. The matrix corresponds to a discretized blurring operator associated with a spatially invariant Gaussian point-spread function (PSF). It is thus block Toeplitz with Toeplitz blocks (BTTB). We used block size 16, which yields a BTTB matrix of dimension 256×256. The parameters used for constructing the matrix are `nband` $= 7$ and $\sigma = 1.0$. The condition number of A is $\kappa(A) \approx 4 \times 10^3$. We used $\tau = 10$ and $k = 2$ to construct a block circulant matrix C_e with an exponential frequency response. It is clear from Figure 10 that the smallest eigenvalues of $C_e^{\mathrm{T}} A C_e$ (marked with '+') overlap with the smallest eigenvalues of A extremely well, as expected. However, the reduction of the largest eigenvalues is not as dramatic as in our earlier example. Only one order of magnitude reduction is achieved by C_e. Since this matrix is BTTB, we can use Chan and Olkin's technique [2] to construct a BCCB matrix, C_t, that minimizes $\|C_t - A\|_{\mathrm{F}}$. The Chan and Olkin's construction appears to be very effective in this case. Most of the eigenvalues of $C_t^{-1} A$ lie within $[0.3, 3]$. Many of them cluster near 1.0. One outlier is found near 10.

Figure 11 shows the performance improvement of the CG iteration after block circulant preconditioners are used. We plotted the relative error norm defined by (10) against the iteration number. Without using any preconditioner, the relative error of the solution is reduced by less than three orders of magnitude in 100 CG iterations. Using the block circulant conditioner C_e constructed by our matrix-free technique, we attained more than five orders of

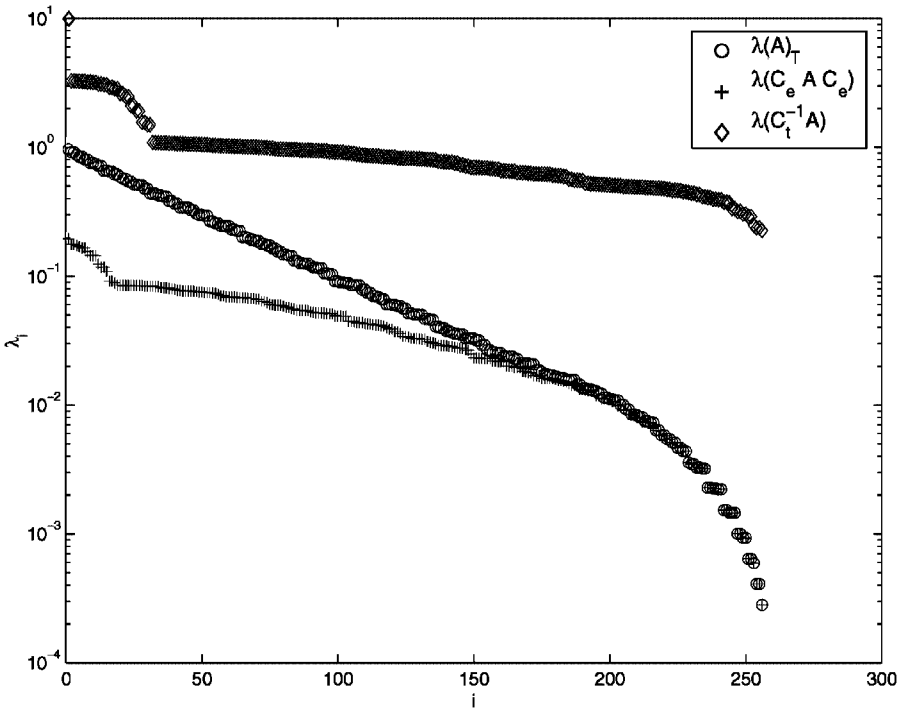

Figure 10. The spectrum of A and the preconditioned matrix.

magnitude reduction in relative error. However, in this example, our matrix-free construction is outperformed by Olkin and Chan's BCCB preconditioner, which yields an error reduction of nearly six orders of magnitude in less than 30 iterations. This example indicates that when A is BTTB, our BCCB preconditioner, which is not optimal in terms of $\|C_e^{\mathrm{T}}C_e - A\|_{\mathrm{F}}$, may not work as well as the Olkin and Chan's construction. However, our construction has the advantage of not making explicit use of the matrix elements of A, a feature that becomes important for the next example.

The matrix A used in our final example arises from the problem of reconstructing a 2-D image x defined on a 17×17 sampling grid from a number of 1-D projections represented by a set of vectors $\{b_i\}$. For the purpose of this paper, let us assume the projections are noise free. The projection associated with a particular projection angle θ_i can be described, in a continuous formulation, by

$$g_i(s) = \mathscr{P}_{\theta_i} x(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(u,v)\delta(s - u\cos\theta_i - v\sin\theta_i)\,\mathrm{d}u\,\mathrm{d}v \tag{11}$$

where $\delta(\cdot)$ is the standard Dirac delta function. The projection operation can be discretized to yield

$$b_i = P_{\theta_i} x$$

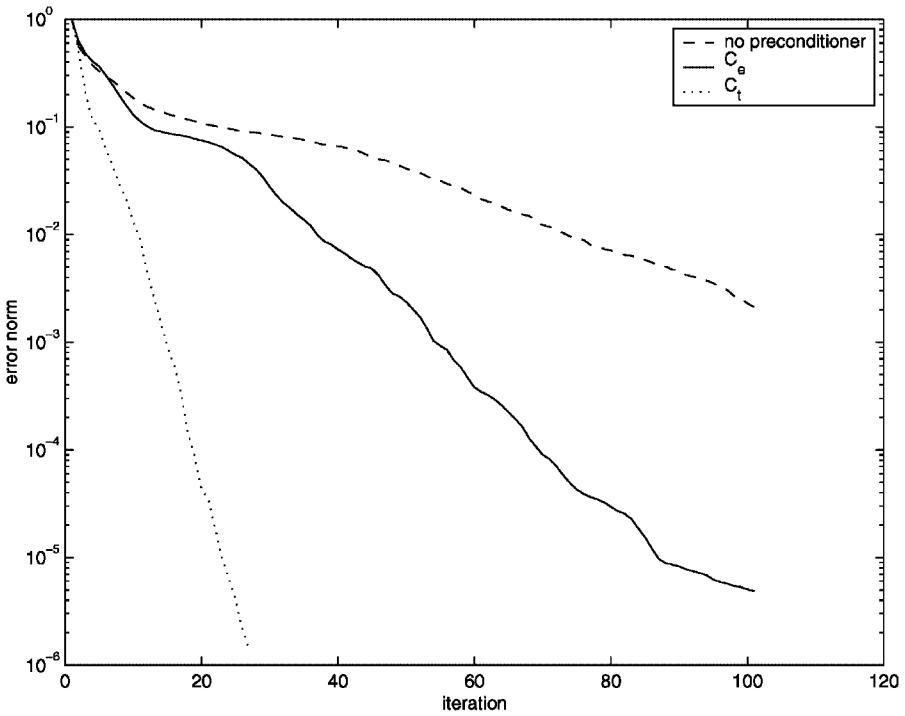

Figure 11. The convergence history of CG.

where b_i and x are vector representations of the uniformly sampled $g(s)$ and $x(u,v)$, respectively, and the non-zero structure of P_{θ_i} depends on the interpolation scheme used to approximate (11).

If we let

$$P = [P_{\theta_1}^{\mathrm{T}} \; P_{\theta_2}^{\mathrm{T}} \cdots P_{\theta_m}^{\mathrm{T}}]^{\mathrm{T}} \quad \text{and} \quad b = [b_1^{\mathrm{T}} \; b_2^{\mathrm{T}} \cdots b_m^{\mathrm{T}}]^{\mathrm{T}}$$

the image reconstruction problem can be formulated as

$$\min_x \|Px - b\|$$

where $\|\cdot\|$ denotes the standard Euclidean norm. The solution of this minimization problem clearly satisfies the normal equation

$$P^{\mathrm{T}}Px = P^{\mathrm{T}}b$$

The transpose of P is often known as the back projection operator, and the direct back projection $\hat{b} \leftarrow P^{\mathrm{T}}b$ typically produces a blurry image. The matrix $A = P^{\mathrm{T}}P$ plays the role of a spatially variant PSF. As mentioned in Section 1, for large size images, it is very costly to compute and store $A = P^{\mathrm{T}}P$ explicitly. This is not necessary when an iterative solver is applied to (4). However, in order to compare the performance of our matrix-free construction of a block circulant preconditioner C_e with that of C_t formed by Olkin and Chan's technique,

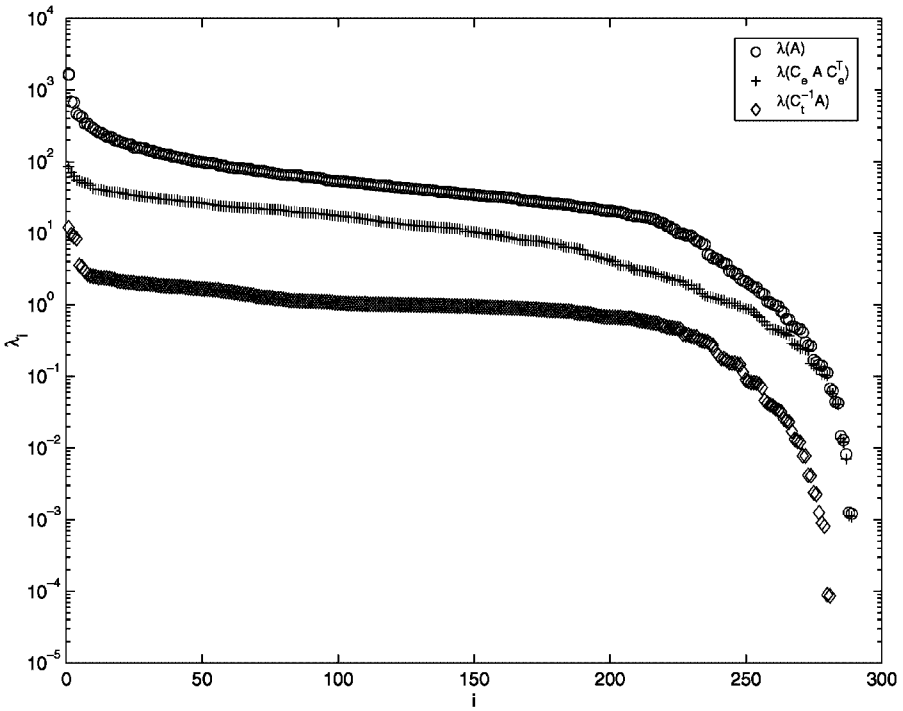

Figure 12. The spectrum of A, $C_e^{\mathrm{T}}AC_e$ and $C_t^{-1}A$.

we built A explicitly by apply P and P^{T} to point sources defined on the sampling grid. Note that A does not have a BTTB structure. However, it is well known that eigenvectors of A can be well represented by a small number of harmonics [22, 23]. We will refer readers to References [15, 24] for additional properties of the projection and back projection operator. The 1-D projections are generated from 100 evenly distributed angles between $[0, 360°]$. The condition number of A is roughly 10^6. We used $\tau = 10$ and $k = 2$ and a 13×13 (polynomial of degree 6) impulse response to construct C_e. The spectrum of A, $C_e^{\mathrm{T}}AC_e$ and $C_t^{-1}A$ are displayed in Figure 12. Notice that the smallest eigenvalues of A and $C_e^{\mathrm{T}}AC_e$ again show significant overlap. The largest eigenvalues of $C_e^{\mathrm{T}}AC_e$ are roughly two orders of magnitude smaller than those of A. This leads to a two orders of magnitude reduction of the condition number. The condition number of $C_t^{-1}A$ remains 10^6. This is partly due to a few eigenvalue outliers at both ends of the spectrum.

In Figure 13, we compare the convergence history of PCG associated with different block circulant preconditioners are used. If we use $X(i, j)$ to denote the intensity of the image at pixel (i, j), $i, j = 1, 2, \ldots, 17$, then the image used in this experiment is set up such that

$$X(i,j) = \begin{cases} 1 & \sqrt{(i-8)^2 + (j-8)^2} < 8 \\ 0 & \text{otherwise} \end{cases}$$

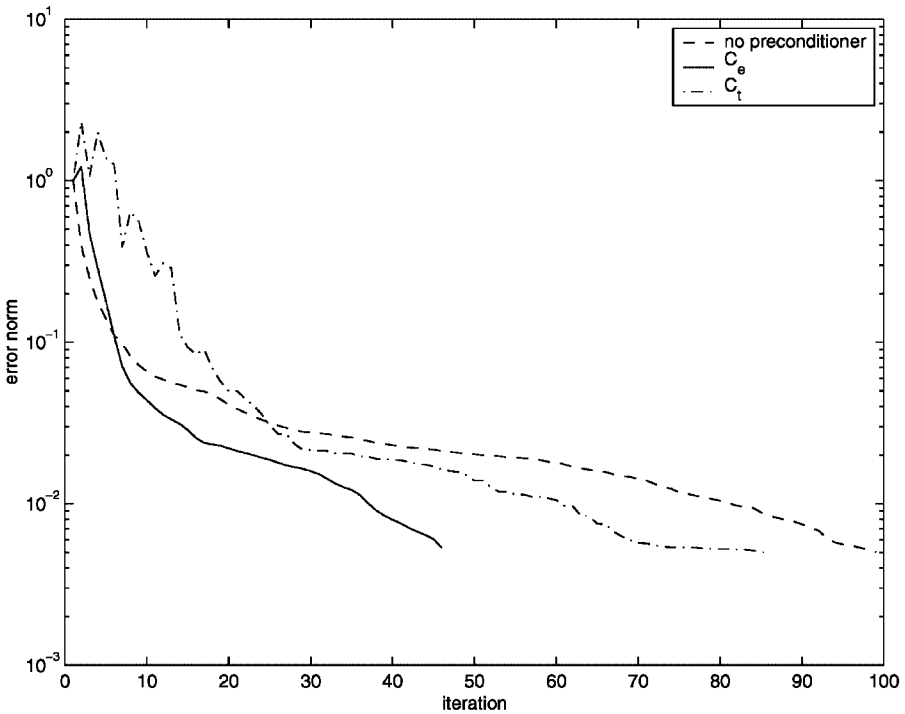

Figure 13. The convergence history of PCG on the image reconstruction problem.

The exact solution x_e is constructed by stacking columns of X on top of one another. The right-hand side b is created by multiplying A with x_e. The CG convergence tolerance is set to $\text{tol} = 5 \times 10^{-3}$. We plotted the relative error defined in (10) against the iteration number of PCG. It is easy to see that without using a preconditioner, nearly 100 iterations are required to reach the desired accuracy. The matrix-free block circulant preconditioner C_e based on the exponential response discussed above helped to reduce the number of iteration by 50%. There appears to be little gain in performance when the block circulant preconditioner C_t constructed by Chan and Olkin's technique is used. This observation is not surprising. When A is far from BTTB, C_t does not provide an optimal solution to $\min \|C_t - A\|_{\mathrm{F}}$ in general.

5. CONCLUDING REMARKS

We have presented a matrix-free framework for constructing circulant and block circulant preconditioners suitable for accelerating the convergence of iterative solution of linear systems arising from the area of image reconstruction [15]. The main characteristic of these systems is that the eigenvectors of the coefficient matrix A can be well represented by a small number of Fourier modes. The close connection between the eigenvectors and the Fourier modes indicates that spectral properties of the linear system (1) can be altered in a systematic fashion by multiplying (1) with a circulant or block circulant matrix C that has an appropriate frequency

response (spectrum). We provided an example of how to devise such an frequency response. The exponential frequency response we proposed has the effect of making the smallest eigenvalues of $C^{\mathrm{T}}AC$ overlap with those of A extremely well. The largest eigenvalues of $C^{\mathrm{T}}AC$ can be made much smaller than those of A. As a result, the condition number of (1) can be reduced significantly, and eigenvalues of the preconditioned system can be squeezed into a much smaller interval.

Our original goal was to develop preconditioners for systems in which the task of explicitly forming A is too costly, a situation that occurs in image reconstruction applications [25]. However, the circulant and block circulant preconditioners developed here can also be used for Toeplitz and BTTB systems. Our preliminary numerical results indicated the performance of our preconditioner is comparable to, and sometimes is even better than that rendered by the Chan and Olkin's preconditioner [1, 2].

We would like to point out that the exponential response presented here is merely one way to specify the frequency response of a circulant or block circulant preconditioner. Other possibilities exist also. For example, if reliable estimation of the spectrum of A is available, one may design a frequency response that interpolates the inverse of the approximate eigenvalues of A.

When (1) is ill-posed, we must take special measures to prevent the unwanted eigenvectors of A to be included in the solution vector because these vectors are often contaminated by noise. If (1) arises from a discretized integral operator, the unwanted eigenvector are those associated with the smallest eigenvalues of A [26]. A simple spectrum squeezing technique may allow some of these components to converge rapidly, resulting in a solution completely corrupted by noise. A potential remedy is to combine the circulant and block circulant preconditioner proposed in this paper with a low-pass filter to prevent the noisy components from entering x. The low-pass filter essentially plays the role of regularization. This subject will be the focus of our future research.

ACKNOWLEDGEMENTS

We would like to thank Drs Daniela Calvetti and Eldad Haber for helpful discussions on preconditioning ill-posed problems at the 2001 International Conference on Preconditioning in Tahoe. We would also like to thank the anonymous referee for providing useful comments that helped to improve the quality of the paper. This work was supported in part by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098. It was also supported in part by the National Institute of Health grant R01 GM60635. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.

REFERENCES

1. Chan TF. An optimal circulant preconditioner for Toeplitz systems. *SIAM Journal on Statistical and Scientific Computing* 1988; **9**:766–771.
2. Chan TF, Olkin J. Preconditioners for Toeplitz-block matrices. *Numerical Algorithms* 1994; **6**:89–101.
3. Chan RH, Yueng MC. Circulant preconditioners constructed from kernels. *SIAM Journal on Numerical Analysis* 1992; **29**:1093–1103.
4. Ku TK, Kuo CJ. Design and analysis of Toeplitz preconditioners. *IEEE Transactions on Signal Processing* 1992; **40**:129–141.
5. Huckle T. Circulant and skew-circulant matrices for solving Toeplitz matrix problems. In *Proceedings of the Copper Mountain Conference on Iterative Methods*, Copper Mountain, CO, 1990.

6. Tismenetsky M. A decomposition of Toeplitz matrices and optimal circulant preconditioning. *Linear Algebra and its Applications* 1991; **154–156**:105–121.
7. Tyrtyshnikov E. Optimal and superoptimal circulant preconditioners. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**:459–473.
8. Nagy JG, O'Leary DP. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing* 1998; **19**:1063–1082.
9. Chan RH. The spectrum of a family of circulant preconditioned Toeplitz systems. *SIAM Journal on Numerical Analysis* 1989; **26**:503–506.
10. Chan RH, Strang G. Toeplitz equations by conjugate gradients with circulant preconditioner. *SIAM Journal on Statistical and Scientific Computing* 1989; **10**:104–119.
11. Chan RH, Yueng MC. Circulant preconditioners for Toeplitz matrices with positive continuous generating functions. *Mathematics of Computation* 1992; **58**:233–240.
12. Chan RH, Nagy JG, Plemmons RJ. FFT-based preconditioners for Toeplitz-block least squares problems. *SIAM Journal on Numerical Analysis* 1993; **30**:1740–1768.
13. Chan RH, Nagy JG, Plemmons RJ. Circulant preconditioned Toeplitz least squares iterations. *SIAM Journal on Matrix Analysis and Applications* 1994; **15**:80–97.
14. Chan RH, Tang PP. Constrained Minmax approximation and optimal preconditioners for Toeplitz matrices. *Numerical Algorithms* 1993; **5**:353–364.
15. Herman GT (ed.). *Image Reconstruction from Projections, Implementation and Applications*. Springer: Berlin, 1979.
16. Hansen PC. Regularization tools. *Numerical Algorithms* 1994; **6**:1–35.
17. Phillips DL. A technique for the numerical solution of certain integral equations. *Journal of the ACM* 1962; **9**:84–97.
18. Davis PJ. *Circulant Matrices*. John-Wiley: New York, NY, 1979.
19. Oppenheim AV, Schafer RW. *Discrete-Time Signal Processing* (2nd edn). Prentice-Hall: Englewood Cliffs, NJ, 1989.
20. Parks TW, Burrus CS. *Digital Filter Design*. John Wiley & Sons, Inc.: New York, NY, 1987.
21. Cheney EW. *Introduction to Approximation Theory*. McGraw-Hill: New York, 1966.
22. Davison ME. A singular value decomposition for the Radon transform in *n*-dimensional Euclidean space. *Numerical Functional Analysis and Optimization* 1981; **3**:321–340.
23. Louis AK. Orthogonal function series expansions and the null space of the Radon transform. *SIAM Journal on Mathematical Analysis* 1984; **15**:621–633.
24. Deans ST. *The Radon Transform and Some of Its Applications*. Krieger Publishing Company: Malabar, FL, 1993.
25. Penczek PA, Radermacher M, Frank J. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy* 1992; **40**:33–53.
26. Hanke M, Nagy JG, Plemmons RJ. Preconditioned iterative regularization. In *Numerical Linear Algebra*, Reichel L, Ruttan A, Varga RS (eds). de Gruyter: Berlin, 1993; 141–163.

